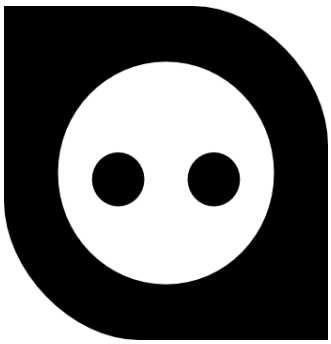


OBFUCIA | Documentation

Programmer's Manual

Contents

Scope Statement

Project Justification

Roger West, our boss, has requested the development of this project to help a client develop a C# internal standalone application that allows text and image file types to be hidden in a targeted media files. This will allow others with the proper security clearance to discreetly obfuscate and send sensitive media without being detected.

High-Level Requirements

- OBFUCIA will run on a Windows 10 computer using C#.
- OBFUCIA will run using UWP (Universal Windows Platform).
- End users should be able to upload text and image media from the local computer or be able to take picture of images using a webcam.
- Allow the option of an image be hidden in a target image file.
- Allow the option of ASCII text be hidden in a target image file.
- An option to save processed media to a users local machine.

Limits and Exclusions

Network functionality should be limited to a user's local host. End users will not be provided a webcam and must provide their own.

The following features can be added ONLY if time permits:

- Hiding ASCII text in a target video file.
- Hiding ASCII in a target audio file.
- Hiding audio in a target image file.
- Additional security such as permissions to data to certain users.

Definitions, acronyms, and abbreviations

UWP – Universal Windows Platform

Stego – Short for Steganography

RGB – Red, Green, and Blue

LSB – Least Significant Bit

References

SimpleImageEditing
Library to simplify pixel manipulation with UWP SoftwareBitmap objects. Created by: Trevor Baron. https://github.com/TrevorDev/SimpleImageEditing

General Description

Project Perspective

This software was built so that future clients could be able to encrypt and decrypt sensitive material into images without detection. With privacy being a primary concern especially online and in the IT industry, Obfucia provides a lightweight steganography platform solution to users using the latest versions of Windows. Obfucia's straight forward and easy to use interface is easy to learn and can process all images regarding of size and shape. Using Obfucia proprietary algorithms only images encrypted using Obfucia can only be deciphered using our software.

Product functions

Obfucia allows users to select images on their computer and encrypt and hide their payload into a target image. Users also have the option to hide written text into an image if preferred. The Obfucia software was designed so that additional encryption and decryption methods could be expanded later upon user request.

User Characteristics

Obfucia was primarily built for our business clients who needed a security solution. However, anyone who is interested in steganography or simply wants to use our software to hide and decrypt images will be able to use Obfucia with ease.

General Constraints

- Obfucia works only on the Windows 10 operating system since the program uses Universal Windows Platform (UWP).
- The OBFUCIA program does not work over a network
- Images must be processed as .PNG files
- When decrypting an image from an image, Image will not return to 1:1 ratio unless both images used are the same pixel size.
- When decrypting an image from an image, There will be some quality loss from the hidden photo.

Assumptions and Dependencies

- We assume that users use Windows 10 Operating Version 1607 or later.
- We assume users use .PNG images.

Specific Requirements

Encrypting Data

1.0.0 This program encrypts a user's chosen data into a user-selected image, and be able to retrieve said data after.

1.1.0 The user-selected image must be a .PNG file-type.

1.2.0 Allowable data to be encrypted is restricted to ASCII text and .PNG images.

Encrypting Images

2.0.0 User should be allowed to choose an origin image to encrypt into the chosen target image.

2.1.0 Origin and target image should be of .PNG file-type

2.2.0 Origin image should be of satisfactory size, so as to be able to properly encrypt into the target image

2.2.1 Go to ImagetoImage class and initialize objects. The origin images are encrypted using the 2 LSB's of the RGB Pixel's bytes.

2.2.2 Start the image convert and then return new bitmap image at the end.

2.2.3 Create the two-bits to hide in the new image.

2.2.4. Initialize method to clear two bits and adds the newly gained bits.

2.3.0 Target image should remain relatively unaffected by the encryption, visually appearing indistinguishable.

Encrypting Text

3.0.0 User should be allowed to input ASCII text to encrypt into the chosen target image.

3.1.0 ASCII text length should be short enough to properly encrypt into the target image.

3.1.1 ASCII text is encrypted using the first-LSB of the RGB Pixel's Bytes.

3.2.0 ASCII characters encrypted should only consist of 1 byte of data per character.

3.3.0 Target image should remain relatively unaffected by the encryption, visually appearing indistinguishable

Decrypting Data

4.0.0 User should be allowed to decrypt any image that was encrypted using Obfucia software.

4.1.0 Image to be decrypted should not have been changed in any way from how Obfucia output it.

4.2.0 User should be able to select to retrieve ASCII text or a .PNG image from the target image to be decrypted.

4.2.1 The image to be decrypted should have been encrypted using the same encryption pattern. To retrieve text, text must have been encrypted, and the same for images.

4.3.0 Retrieved data is not saved to the users local machine, and is only viewable inside Obfucia.

4.4.0 Image to be decrypted will be un-altered in this decryption process, and may be used again.

Camera

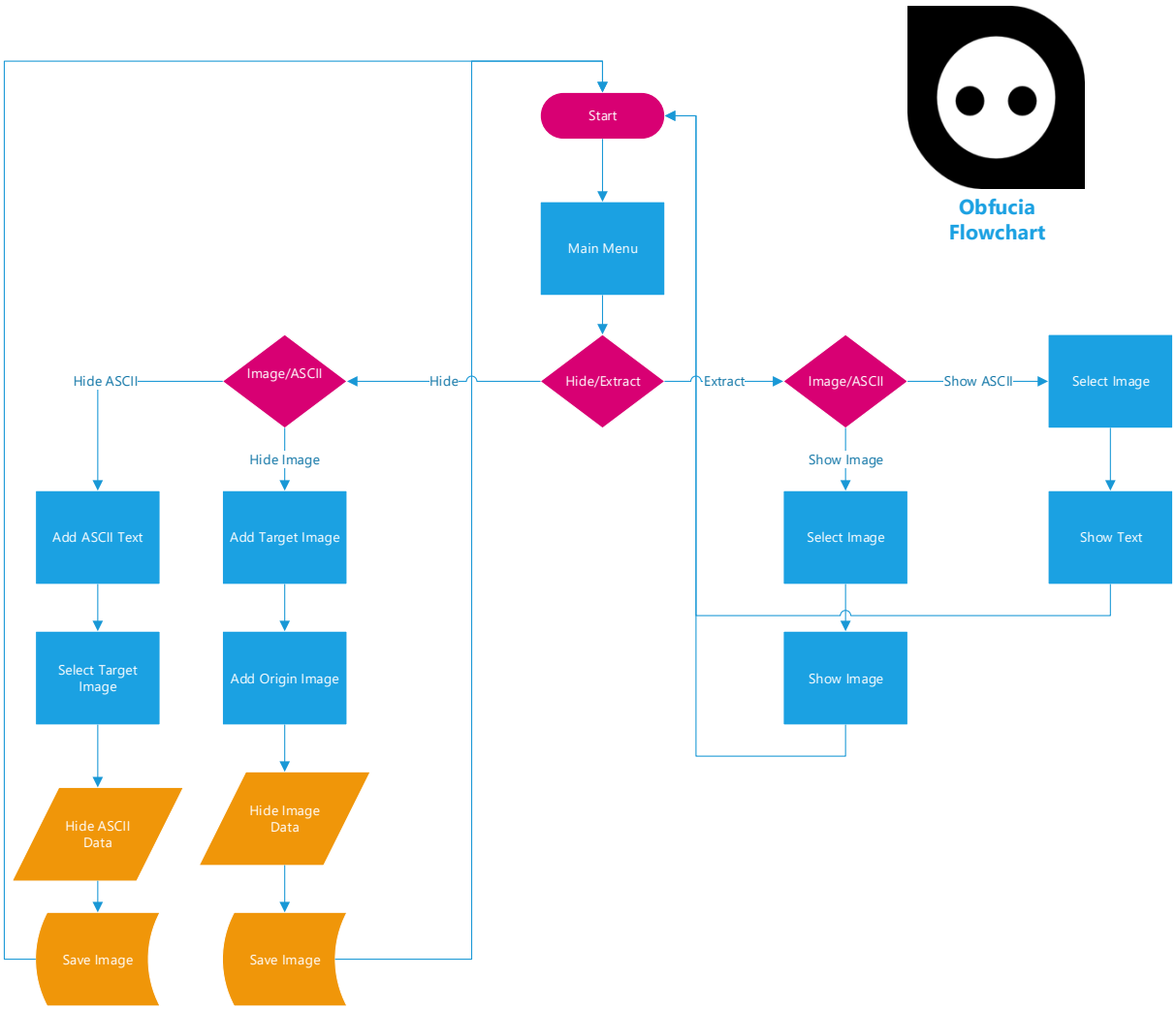
5.0.0 Users should be able to use the webcam to create origin or target images

5.1.0 The webcam image must be a .PNG file-type.

5.1.1 The webcam image must load and display correctly.

Design Documentation

Activity Diagram



Pseudocode

The following pseudocode goes over the logic of the hiding and deciphering process of the steganography.

Hide an image in an image (ImageToImage)

Initialize both origin and target Bgra8 SoftwareBitmaps and start image convert.
Proceed to source method
Retrieve the width and height of the origin SoftwareBitmap
Create a new bitmap from origin bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.
    Loop through with and height of origin SoftwareBitmap
        Get the individual pixel
        Get the individual Red Green and Blue values
        Divide the individual RGB values by 64 to create the integer 2-bit patterns
        Set the new integer 2-bit pattern value to the individual pixel.
    End loop
Return the modified pixels
Proceed to hide method
Retrieve the width and height of the origin SoftwareBitmap again
Create a new bitmap from target bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.
    Loop through with and height of origin SoftwareBitmap
        Get the individual pixel from Bitmap editor and from modified pixels from previous method
        Get the individual Red Green and Blue values and assign values from BitmapEditor and modified pixels.
        Clear 2bits from each RGB value and then add the modified pixels to them
        Set the new pixel value to the individual pixel.
    End loop
Return new pixel values

Hide ASCII to an image (TextToImageHider)

Initialize target Bgra8 SoftwareBitmap and the original string.
Retrieve the width and height of the target SoftwareBitmap
Create a new bitmap from target bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.

Loop through with and height of target SoftwareBitmap
Get the individual pixel
Clear out the LSB
Loop through each pixel color value
Check to see if we need a new ascii value to process
If we've gone through the length of the string
fill any remaining pixels with 0's in their LSB's
Else
get character value then increment index to next character.
Switch through every RGB value and hide the string
Increment pixelIndex
End loop
End loop
Return new pixel values

Extract an image in an image (ExtractImage)

Initialize extractedImage SoftwareBitmap and start image convert.
Proceed to extract method
Retrieve the width and height of the extractedImage SoftwareBitmap
Create a new bitmap from extractedImage bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.
    Loop through with and height of exxtractedImage SoftwareBitmap
        Get the individual pixel
        Get the individual Red Green and Blue values
        Mod the individual RGB values by 4 to extract the integer 2-bit patterns
        Set the new integer 2-bit pattern value to the individual pixel.
    End loop
Return the modified pixels
Proceed to reconstruct method
Retrieve the width and height of the extractedImage SoftwareBitmap again
Create a new bitmap from extractedImage bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.
    Loop through with and height of extractedImage SoftwareBitmap
        Get the individual pixel from Bitmap editor and from modified pixels from previous method
        Get the individual Red Green and Blue values and assign values from BitmapEditor and modified pixels.
        Recreate pixel by multiplying each modified RGB value by 64
        Set the new pixel value to the individual pixel.
    End loop
Return new pixel values

Extract ASCII from an image (TextToImageHider)

Initialize target Bgra8 SoftwareBitmap
Retrieve the width and height of the target SoftwareBitmap
Assign image bitmap from target bitmap with Bgra8 format and create a new SoftwareBitmapEditor to edit pixels.
    Loop through with and height of image SoftwareBitmap
        Get the individual pixel
        Loop through each pixel color value
            Switch through every RGB value and retrieve the character value
            Increment pixelIndex
            Check to see if we need a new ascii value to process
                Flip the bit values
            If the character value is 0
                Return the output text
            Else
                Add character value to the output text
        End Loop
    End loop
Return output text

Testing Documentation

Test Case #	Requirement Tested	Rationale	Input(s)	Expected Output	Passed?
1	1.0.0	User should be allowed to see the Encryption menu	User Clicks Icon for Encryption	User is shown the page for Encryption	Y
2	1.0.0	User should be allowed to see the decryption menu	User clicks Icon for decryption	User is shown the page for decryption	y
3	1.1.0	User should only be allowed to use .PNG images	User selects "Use local image" button	User should only be given the choice of selecting .PNG images	y
4	3.1.0	User's text should not be encrypted if it is too large to fit inside the target image	User's inputs text that will not fit in a target image	Background validation occurs, and refuses to let the user progress with encryption	y
5	3.3.0	Encrypted images should appear visually the same	User encrypts an image and saves their local copy	Outputted image, which when visually inspected, appears same as target image was.	y
6	2.1.0	Images to be used for encryption should be of .PNG type	User selects "Select image..." icon	Only .PNG images are allowed to be chosen	y

7	2.2.0	Image to be used for encryption should be small enough to fit inside the target	User selects a larger image to try and encrypt	Background Validation will reject users image and prevent user from progressing	Y
8	4.2.0	User should be able to decrypt images that contain data for ASCII	User selects an image that contains encrypted text	Decryption will return the string that was hidden	Y
9	4.2.0	User should be able to decrypt images that contain data for other images	User selects an image that contains an encrypted .PNG image	Decryption will return a rough equivalency of the hidden image.	Y
10	4.3.0	For security purposes, users should only be able to view decrypted data in the final decryption screen of Obfucia	User decrypts data and selects "Done" button	Decrypted data is not saved to drive, and disappears from screen	Y
11	4.4.0	Images with hidden data should be unaltered by the process of retrieving said data	User decrypts information from an image	Image with information hidden is unaltered on users drive	Y
12	5.0.0	Users should be able to use a webcam to take an image for use	User selects the "Take image with webcam" button	Webcam on users machine is used to take an image within Obfucia	Y